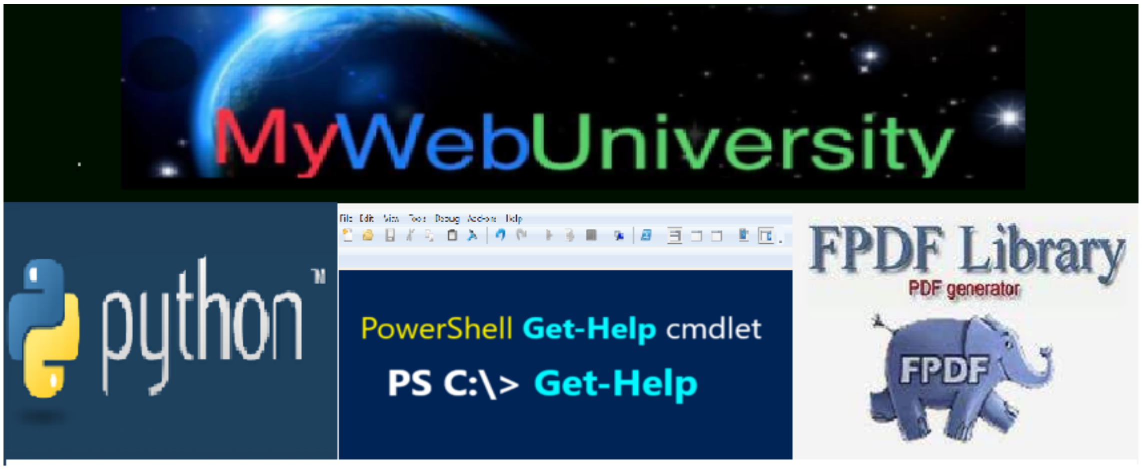

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

# *Windows PowerShell Get-Help on Cmdlet 'Export-ModuleMember'*

***PS:\>Get-HELP Export-ModuleMember -Full***

NAME

Export-ModuleMember

SYNOPSIS

Specifies the module members that are exported.

SYNTAX

Export-ModuleMember [[-Function] <System.String[]>] [-Alias <System.String[]>] [-Cmdlet <System.String[]>] [-Variable <System.String[]>] [<CommonParameters>]

DESCRIPTION

The `Export-ModuleMember` cmdlet specifies the module members that are exported from a script module (`.psm1`) file, or from a dynamic module created by using the

`New-Module` cmdlet. Module members include cmdlets, functions, variables, and aliases. This cmdlet can be used only in a script module file or a dynamic module.

If a script module does not include an `Export-ModuleMember` command, the functions and aliases in the script module are exported, but the variables are not. When a script module includes `Export-ModuleMember` commands, only the members specified in the `Export-ModuleMember` commands are exported. You can also use

`Export-ModuleMember` to suppress or export members that the script module imports from other modules.

An `Export-ModuleMember` command is optional, but it is a best practice. Even if the command confirms the default values, it demonstrates the intention of the module

author.

PARAMETERS

-Alias <System.String[]>

Specifies the aliases that are exported from the script module file. Enter the alias names. Wildcard characters are permitted.

| | |
|---|---|
| Required? | false |
| Position? | named |
| Default value | None |
| Accept pipeline input? | True (ByPropertyName) |
| Accept wildcard characters? | true |

-Cmdlet <System.String[]>

Specifies the cmdlets that are exported from the script module file. Enter the cmdlet names. Wildcard characters are permitted.

You cannot create cmdlets in a script module file, but you can import cmdlets from a binary module into a script module and re-export them from the script module.

| | |
|---|---|
| Required? | false |
| Position? | named |
| Default value | None |
| Accept pipeline input? | True (ByPropertyName) |
| Accept wildcard characters? | true |

-Function <System.String[]>

Specifies the functions that are exported from the script module file. Enter the function names. Wildcard characters are permitted. You can also pipe function

name strings to `Export-ModuleMember`.

Required?                    false

Position?                    0

Default value                None

Accept pipeline input?       True (ByPropertyName, ByValue)

Accept wildcard characters?  true

-Variable <System.String[]>

Specifies the variables that are exported from the script module file. Enter the variable names, without a dollar sign character (`$`). Wildcard characters are

permitted.

Required?                    false

Position?                    named

Default value                None

Accept pipeline input?       True (ByPropertyName)

Accept wildcard characters?  true

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.String

You can pipe function name strings to this cmdlet.

OUTPUTS

None

This cmdlet returns no output.

NOTES

- To exclude a member from the list of exported members, add an `Export-ModuleMember` command that lists all other members but omits the member that you want to

exclude.

-- Example 1: Export functions and aliases in a script module --

```
Export-ModuleMember -Function * -Alias *
```

This command exports all the functions and aliases defined in the script module.

------- Example 2: Export specific aliases and functions -------

```
Export-ModuleMember -Function Get-Test, New-Test, Start-Test -Alias gtt, ntt, stt
```

This command exports three aliases and three functions defined in the script module.

You can use this command format to specify the names of module members.

----------------- Example 3: Export no members -----------------

```
Export-ModuleMember
```

This command specifies that no members defined in the script module are exported.

This command prevents the module members from being exported, but it does not hide the members. Users can read and copy module members or use the call operator (`&`)

to invoke module members that are not exported.

------------ Example 4: Export a specific variable ------------

```
Export-ModuleMember -Variable increment
```

This command exports only the `$increment` variable from the script module. No other members are exported.

If you want to export a variable, in addition to exporting the functions in a module, the `Export-ModuleMember` command must include the names of all of the functions
and the name of the variable.

------------- Example 5: Multiple export commands -------------

```
# From TestModule.psm1
function New-Test
{
    Write-Output 'I am New-Test function'
}
Export-ModuleMember -Function New-Test

function Validate-Test
{
    Write-Output 'I am Validate-Test function'
}
function Start-Test
{
    Write-Output 'I am Start-Test function'
}
Set-Alias stt Start-Test
Export-ModuleMember -Function Start-Test -Alias stt
```

These commands show how multiple `Export-ModuleMember` commands are interpreted in a script module (`.psm1`) file.

These commands create three functions and one alias, and then they export two of the functions and the alias.

Without the `Export-ModuleMember` commands, all three of the functions and the alias would be exported. With the `Export-ModuleMember` commands, only the `New-Test`

and `Start-Test` functions and the `STT` alias are exported.

-------- Example 6: Export members in a dynamic module --------

```
New-Module -Script {function SayHello {"Hello!"}; Set-Alias Hi SayHello; Export-ModuleMember -Alias Hi -Function SayHello}
```

This command shows how to use `Export-ModuleMember` in a dynamic module that is created by using the `New-Module` cmdlet.

In this example, `Export-ModuleMember` is used to export both the `Hi` alias and the `SayHello` function in the dynamic module.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/microsoft.powershell.core/export-modulemember?view=powershell-5.1&WT.mc_id=ps-gethelp

Get-Module

Import-Module

Remove-Module

about_Modules